System and method for non-programmers to dynamically manage multiple sets of XML document data

Abstract

A system for manipulating, updating, creating, and displaying data from sources of Extensible Markup Language (XML) documents. The information from one or more XML document sources is structured and reorganized into management records. The management records interact with the dynamic documents to reformat the data into the form desired by the user. Both the management records and dynamic documents are also used to execute operations on the data such as sorts, filters, and logical and mathematical functions. The present invention also include a plurality of unique methods for defining a point of view for viewing the data, a method for assembling the data extracted from the XML documents, a method for reorganizing the data in management records into any form of dynamic documents, and methods for displaying and printing dynamic documents.

Inventors: **VanderDrift, Richard William**; *(Larkspur, CA)*

Correspondence Name and Address:
Richard William VanderDrift
130 Magnolia Avenue
Larkspur
CA
94939
US

Serial No.: **976710**

Series Code: **09**

Filed: **October 15, 2001**

U.S. Current Class: **709/218**; 715/760

U.S. Class at Publication: **709/218**; 345/760

Intern'l Class: G06F 015/16; G09G 005/00

Claims

I claim:

1. A computer implemented method for dynamically transforming and displaying, and optionally entering and updating data from XML data sources, said method comprising the steps of: a. reading a plurality of XML formats; b. defining a plurality of primary record types from components of the XML format that have multiple occurrences or have child components; c. defining a plurality of parent-child relationships between the primary record types from the relationships between the XML format components; d. defining a management record by choosing an XML format or by selecting a plurality of

primary record types for inclusion in the management record; e. defining a dynamic document by choosing an XML format or by selecting a management record type, a plurality of the selected management record type's primary record types, and defining a hierarchical relationship for the selected primary record types; f. creating and updating primary record instances from XML documents; g. displaying data from the primary record instances using the dynamic document instances; h. creating an XML document definition for the dynamic document; and i. writing an XML document for the dymanic document instance.

Deleted: its

2. The method of claim 1, wherein the step of defining a dynamic document includes any of the steps of: a. generating management record pointer families for each management record definition, each management record pointer family comprising a data structure defining a group of related primary record types within the management record, the management record pointer families generated by performing the steps of: i. retrieving a management record; ii. retrieving a unique primary record type of the management record; iii. using the unique primary record type as the current primary record type; iv. creating a family pointer type for a current primary record type; v. retrieving a set of primary record definitions related to the current primary record type; vi. creating a child record type to the family pointer for each primary record type retrieved in substep 5) that has a one to many relationship with the current primary record type: and vii. for each primary record type retrieved in substep 5) that has a relationship with the current primary record type, repeating substeps 4-7 with the primary record type retrieved as the current primary record type; b. constructing the management record pointer instances; c. creating the dynamic document instances for the management record pointer instance.

Deleted: the

3. The method of claim 1, wherein the step of defining a primary record further comprises the step of identifying a plurality of the primary record's fields as the unique identifying fields.

Deleted: its

Deleted:

4. The method of claim 1, wherein the step of defining a management record further comprises the step of identifying one of the selected primary record types as the unique identifying primary record for the management record.

5. The method of claim 1, wherein the step of defining a dynamic document includes any of the steps of: a. defining a function that uses the data of the data source; b. defining a filter that operates on the data of the data source, the filter specifying a set of data selection criteria; and C. defining a sort that operates on the data of the data source.

6. The method of claim 1, wherein the step of creating and updating primary records from XML documents, further comprise the steps of: a. identify the primary record that corresponds to the XML format component; b. grouping the XML the component instances by the primary record's key fields into a temporary table; and c. inserting a primary record instance for each primary record type key field grouped by temporary table row.

Deleted: their

7. The method of claim 1, further comprising the step of executing a function on the

management records.

8. The method of claim 1, wherein the method executes user input sorts on the data in the dynamic documents before the step of displaying the data.

9. The method of claim 1, wherein the step of displaying data in the primary records using the dynamic document instance comprises the steps of: a. sending the data in an organized format to a screen forms display tool; b. placing values into proper fields; and c. using display rules entered by the user to control the appearance of a set of display panes, and positioning the format of fields within the primary record type.

10. The method of claim 1, further comprising the step of executing a filter on a dynamic document, a filter execution corresponding to selecting pane rows, the filter specifying a set of data selection criteria.

11. The method of claim 10, wherein the step of executing a filter on the dynamic documents further comprises the steps of: a. reading the dynamic document; b. reading a filter for a dynamic document definition; c. comparing filter selection criteria to each dynamic document pane row; d. displaying dynamic document pane rows that meet the filter's selection criteria; e. determining whether the filter is a delete type filter; f. deleting dynamic document pointers for pane rows that do not meet the filter's selection criteria and g. deleting the dynamic document pointer's children dynamic document pointers in child panes.

12. The method of claim 1, further comprising the step of executing a function on the management records.

13. The method of claim 12, wherein the step of executing a function further comprises the steps of: a. retrieving the dynamic document definition and a dynamic document instance; b. retrieving a lowest level unprocessed display pane hierarchy level for the dynamic document definition; c. retrieving a set of functions associated with the display pane; d. executing each of the functions; e. updating the dynamic document; and f. updating a group by accumulator, wherein the group by accumulator stores a result corresponding to a calculation performed upon a group of data fields.

14. The method of claim 2, further comprising the step of executing a filter on the management records, the filter execution corresponding to a data selection, the filter specifying a set of data selection criteria.

15. The method of claim 14, wherein the step of executing a filter further comprises the steps of a. reading the management record pointer instance; b. reading the filter from a management record definition; c. comparing a filter value to an appropriate field of the management record pointer instance; d. displaying the management record pointer instance if the filter value matches the appropriate field of the management record pointer instance; e. determining whether the filter is a delete type filter; and f. deleting any management record pointer instance and its children if it is a delete type filter and the

Deleted: , and the type of maintenance that can be performed

Deleted: the

Deleted: the

Deleted: a data

Deleted: selection

Deleted: the

Deleted: from

Deleted: a

Deleted: value

Deleted: an appropriate field of the

Deleted: the

Deleted: pointer

Deleted: if the filter value matches the appropriate field of the dynamic document

Deleted: and

Deleted: any

Deleted: its

Deleted:

Deleted: if it is a delete type filter and the filter value matches the appropriate field of the dynamic document

filter value matches the appropriate field of the management record pointer instance.

16. The method of claim 2, wherein executing a function further comprises steps of: a. retrieving a management record definition and a management record instance; b. retrieving a lowest level unprocessed management record pointer family for the management record definition; c. retrieving a set of functions associated with the management record pointer family; d. executing each function retrieved; e. updating the management record pointer instances corresponding to the management record pointer family; f. updating the parent management record pointer instances of other management record pointer families that use the function; and g. updating a group by accumulator, wherein the group by accumulator Stores a result corresponding to a calculation performed upon a group of data fields.

Deleted: the step of

Deleted: the

17. The method of claim 16, wherein the method maintains a pass sequence number that is incremented for each level of function nesting in a management record, and executing the function is performed a plurality of times equal in number to the pass sequence number.

Deleted: the step of

18. The method of claim 16, wherein the method maintains a pass sequence number that is incremented for each level of function nesting in a dynamic document, and executing the function is performed a plurality of times equal in number to the pass sequence number.

Deleted: the step of

19. The method of claim 2, wherein the step of defining a dynamic document comprises the step of defining dynamic document pointers, said dynamic document pointers indicating the primary record types to be displayed.

20. The method of claim 19, the step of defining dynamic document pointers further comprises the steps of: a. retrieving a dynamic document definition; b. determining a highest hierarchy level display pane definition not yet processed, the display pane definition defining a display window; c. reading the primary record type and a corresponding management record pointer family for the pane definition; d. determining whether a display pane exists for the primary record type and the corresponding management record pointer family, the display pane corresponding to a display window; and e. if the display pane does not exist, creating a display pane for the primary record type and the corresponding management record pointer family by joining management record pointer instances for this display pane and a display pane one level up hierarchy level.

Deleted: its

Deleted: parent

21. The method of claim 20, further comprising the step of executing filters before and after the step of creating a display pane.

22. The method of claim 20, further comprising the steps of: a. determining whether any child display panes exist; and b. storing a copy of the joining management record pointer instances for each child display pane.

23. The method of claim 1, further comprising the steps of: a. accepting user revisions to the data displayed; b. determining whether the revision is a data revision; c. updating the management record instances if the revision is a data revision; d. determining whether the revision affects a function; e. maintaining and executing the affected functions if the revision affects a function; f. determining whether the revision affects a filter; g. maintaining and executing the affected filters if the revision affects a filter, each filter execution corresponding to a data selection according to a set of data selection criteria; h. determining whether the revision is a sort revision; and i. maintaining and executing the affected sort if the revision is a sort revision,

24. The method of claim 23, wherein the step of maintaining and executing the affected functions if the revision affects a function comprises the steps of: a. retrieving user changes to the function; b. replacing calculated fields used in the function with their function; c. updating primary record types used to point to the changed function; d. updating the management record to point to the changed function; e. reading the management record instance; and f. executing the changed function.

25. The method of claim 23, wherein the step of maintaining and executing the affected functions if the revision affects a function comprises the steps of: a. retrieving user changes to the function; b. replacing calculated fields used in the function with their function; c. updating primary record types used to point to the changed function; d. updating the dynamic document to point to the changed function; e. reading the dynamic document; and f. executing the changed function.

26. The method of claim 23, wherein the step of maintaining and executing the affected filters if the revision affects a filter comprises the steps of: a. retrieving user changes to the filter; b. updating the management record type definition to point to the changed filter; c. reading the management record instance for the filter's management record pointer family; and d. executing the filter.

27. The method of claim 23, wherein the step of maintaining and executing the affected filters if the revision affects a filter comprises the steps of: a. retrieving user changes to the filter; b. updating the dynamic document display pane definition to point to the changed filter, the display pane definition defining a display window; c. reading a display pane row for a display pane associated with the filter; and d. executing the filter.

28. The method of claim 23, wherein the display is updated after the step of maintaining and executing the affected sort if the revision is a sort revision.

29. The method of claim 23 wherein the step of updating the management record instances if the revision is a data revision, comprises the steps of: a. determining whether a record has been changed; b. updating the management record instances for a changed primary record instance if a record has been changed; c. determining whether an record has been added; d. updating the management record instances for a new primary record instance if a record has been added; e. executing the management record filters; f. executing the management record functions; g. updating the dynamic documents; and h.

re-displaying the dynamic documents to show the data revision.

30. The method of claim 29, wherein the substep of updating the management record pointer instances for a changed primary record instance comprises the steps of: a. determining whether field values for a set of related fields are equal; and b. removing a pointer within the management record pointer instance.

31. The method of claim 29, wherein the substep of updating the management record pointer instances for a new primary record instance comprises the steps of: a. retrieving the new primary record instance; b. determining where the new primary record instance is used by reading the primary record type definition and the management record type definition corresponding to the primary record instance; c. determining whether the primary record type is a lead primary record of a management record pointer family; d. creating a management record pointer instance and executing functions and filter associated with the management record pointer instance, if the primary record type is a lead primary record; e. if the primary record type is not a lead primary record, reading the management record pointer instance for the management record pointer family; f. determining whether a field related to the primary record instance is a management record pointer instance; and g. adding a primary record instance pointer to the management record pointer instance.

32. The method of claim 29, wherein the step of executing the management record functions comprises the steps of: a. retrieving the changed primary record instance and a primary record type that corresponds to the primary record instance; b. retrieving the function that uses the changed primary record instance; c. retrieving the management record pointer family of the function that uses the changed primary record instance; d. retrieving the management record pointer instance for the changed primary record instance; e. executing the function; and f. executing functions using the management record pointer instance.

Deleted: its corresponding

33. The method of claim 29, wherein the substep of updating the dynamic documents comprises the steps of: a. identifying changed management record pointer instances and changed primary record instances; b. identifying the dynamic document pointers instances that use the changed fields; c. determining whether any of the changed fields, pointers and filters are used in other nested calculations, and identifying the dynamic document pointer instances that use the other nested calculations; d. executing the filters for the identified dynamic document pointers, each filter execution corresponding to a data selection according to a set of data selection criteria; e. executing the functions for the identified dynamic document pointers; f. re-sorting the records of dynamic document pointers; and g. re-displaying the records of dynamic document pointers.

34. The method of claim 2, wherein the step of writing an XML document for the dynamic document instance when any dynamic document pointer records change: a. identifying the changed primary record instances; b. identifying the dynamic documents pointer records that include changed primary record instances; c. updating the the dynamic documents pointer records affected by the primary record instance changes; d.

Deleted: of its

retrieving the dynamic document definition to XML document type definition maps; e. creating the XML document by copying dynamic document instance data values into the record components; f. copying standard XML document header information and tags into the XML document; and g. handing the XML document and the destination Internet address to a web server.

35. The method of claim 1, wherein multiple individuals can view and maintain identical MRIs through multiple dynamic documents concurrently.

Deleted: the